A Framework for Reverse Engineering DoD Legacy Information Systems

Peter Aiken† Alice Muntz* & Russ Richards†

† Defense Information Systems Agency
Center for Information Management

* Hughes Information Technology Company
Information Systems Technology Laboratory

Abstract

As with most large organizations, the Department of Defense has strategic and economic needs to capitalize on and consolidate existing information systems. This paper reports on a framework to reverse engineer selected legacy information systems in DoD's heterogeneous environment. This approach was developed to recover business rules, domain information, functional requirements, and data architectures, largely in the form of normalized, logical data models. In a pilot study, we are reverse engineering the data from diverse systems--ranging from home grown languages and database management systems developed during the 1950's to those using high order languages and commercial network database management systems. The pilot study is being used to: validate and refine the framework with real-life systems; develop a baseline approach for reverse engineering existing systems; scope and estimate future system re-engineering costs; and determine the economic viability of re-engineering, reverse, and forward engineering efforts.

Introduction

The Department of Defense spends more than $9 billion annually in non-combat information technology development at more then 1,700 DoD Data Centers currently running hundreds of legacy information systems. According to Paul Strassmann, former Director of Defense Information (DDI), in one functional area -- pay -- there are probably more than one hundred pay-related systems servicing the Department [10]. (A recent report to Congress identified more than 50 separate payroll systems [11].) This extensive installed base of existing application software creates two key classes of problems.

- First, the cost of maintaining all of these legacy information systems consumes an enormous portion of total DoD information technology spending. Typical estimates put the cost of software maintenance costs around 60 percent of annual development budgets [5]. Applying this metric to the $9 billion non-maintenance costs for the Department results in an enormous cost to maintain legacy information systems.
- Second, and more importantly, lack of standardized data and data structures across systems results in numerous situations where the Department is unable to obtain information from the data stored in the various databases in existence. Submitting the same query to each of the payroll systems can result in not just multiple answers but in multiple kinds of answers. At times, consolidating the query responses has proven to be an impossible task.

Many of these DoD legacy systems were developed using non-standard development techniques, 'home-grown' database management systems, and obscure programming languages. To address some of these problems, the Defense Information Systems Agency (DISA) and it's Center for Information Management (CIM), the Data Administration Program Management Office (DAPMO) has been charged with developing a data architecture supporting data standardization throughout the Department of Defense. The monetary goals identified for savings to be achieved through the application of corporate information management are significant -- two billion dollars from reduction of information systems (hardware, software, data) costs by FY 1997.

Influenced by information engineering concepts, DAPMO is currently implementing a series of reverse engineering projects designed to contribute to the development of a DoD-wide data architecture. Keeping in mind the need to achieve balance between technical infrastructure correctness and saving money, DAPMO has initiated a Data Processing Systems Modernization (DPSM) program. This effort is designed to provide a refinable and systematic approach to migrate current data, data definitions, and data structures to a modernized, re-engineered environment supporting the DoD data standardization program.

The next section of this paper describes the operational, technical, and administrative complexities in modernizing DoD Data Processing Systems (DPS). Section two describes the impact of the complexity of the legacy environment. Section three describes the DPSM project context. Section four describes the reverse engineering framework, and our approaches to modeling and model management. The last section relates some lessons learned from experience with the framework.

Impact of Legacy Environment Complexity

In modernizing DoD DPS, it is important to deal with three dimensions of complexity: operational, technological, and administrative. Figure 1 characterizes the legacy system operational environment to be reverse engineered in Phase I of the DPSM program. As illustrated, DoD has a heterogeneous legacy operational environment that no single CASE technology can currently address [9].

Operational Complexity

In the DoD systems inventory, there are thousands of DPS. In the past, these systems were built and maintained to satisfy operational requirements of major organizational elements within DoD (Army, Navy, Air Force, Marine Corps, Joint Chiefs of Staff, Unified and Specified Commands, Agencies etc.). Often these major organizational elements would allow subordinate level organizations located worldwide to also maintain separate systems. Thus, hundreds upon hundreds of systems have been propagated producing a condition in which many systems perform duplicate operations.

System	Language(s)	Data Handling System(s)	Operational Environment
Defense Civilian Personnel Data System (DCPDS)	Burroughs assembly Home grown language "Samuel"	Home-grown database management system	Multiple sites using remote access
Defense Civilian Pay System (DCPS)	COBOL	IDMS/R	
Marine Corps Total Force System (MCTFS)	COBOL & Assembly Language Code	VSAM & Adabase	
Composite Health Care System (CHCS)	MUMPS	FILEMAN	
Medical Performance Factors (MEPRS)	MUMPS	FILEMAN	

Figure 1: Example Legacy Environment.

For instance, to pay civilian employees many separate systems were developed; to pay military employees many more; and to manage civilian and military personnel records of both employees and employees of non-appropriated activities still more. To collect and analyze or process data at higher organizational levels, management has to collect data from one or many of the lower levels. This process depends heavily upon subordinate organizations feeding the information upward (often manually). If the information fed upward is electronic, the format has to be meticulously specified at each level because each organizational element has its own technology baseline (discussed below) with its own "standard" format. Consistency and accuracy of data has been difficult to maintain and control in this environment. Even if subordinate organizations are on some of the same systems, physical distribution and slowness in acquiring and incorporating emerging technologies often make operational complexities even worse.

The described operational complexity impacts re-engineering projects on two levels. First, the operational environment is itself a source of physical evidence of what must be captured in the business process and data models. This is especially true when interfaces exist between systems. The interfaces are sources of information, documenting links between physical evidence and data models. Second, even though a specific systems may have been selected to replace a group of systems in a functional domain, it will not be successful unless the system unique requirements of the systems to be replaced are isolated and documented. If these unique requirements are not included in the modernization of the selected system or included in a replacement system, the legacy systems must continue to operate.

Technological Complexity

Baseline data processing systems include obsolete electronics, technology, and systems designs - many 20 to 30 years old and poorly documented. Reliability and maintainability of DoD DPS are pushing the limits of their engineered capabilities. Upgrades through modernization and advanced technology insertion are required to yield enhanced performance and operational capabilities while maintaining operational consistency throughout DoD. Many of these 20 to 30 years old designs cannot be readily adapted to current technology. For instance, one of the information systems being reverse engineered is still using application program managed memory overlays. The need to do this was eliminated years ago with the development of virtual memory. Another example from the data perspective, includes the continued use of flat-file technology. Another of the systems being reverse engineered was originally innovative in its use of flat-files by developing a table-driven approach to separate process from data as relational database technology has done. However, a fixed record length limits the number of fields available to accommodate repeating groups available. These artificial limits were eliminated with the advent of modern database technologies.

As stated previously, the described technological complexity prevents any single reverse engineering CASE tool from addressing all the needs of this heterogeneous environment. This also means that manual analysis is required in combination with commercial tools and supplemented with custom software. This approach has been incorporated in the reverse engineering framework.

Administrative Complexity

In 1979, as an observer of organizational ability to adapt to technological change, Richard Nolan [8] stated "In the stages of control and integration, the dominant forces have to do with organizational discipline and don't relate very closely to technology." The Department of Defense is no exception. Figure 2 illustrates the variety and number of people who require coordination (and briefings) for a single information system. In this example, we were forced to coordinate with 40 people from 11 different organizations. We used these appropriately named "Client Mazes" to track just who said what to whom.

Project Context

Figure 3 depicts the evolutionary process from legacy systems to the DoD target environment. According to guidelines from DoD directives [3, 4], functional officials should eliminate redundant systems and select "migration" systems to provide the essential functionality required in the near-term. The selected migration systems should be modernized and eventually become the "target" systems for the functional areas. Modernization includes separation of data from process, development of universal data structures for reuse throughout the department, and development of shared data that can be used by multiple applications.

The legacy environment largely consists of individual systems with their application-based databases or files. In order for one information system to access data in another, data must be mapped and passed between the two. This can be done by interfacing applications, electronic file exchange, media exchange, etc. One of the goals of the migration process is to separate the application dependent data from the application processes so that they can be directly accessed by any other individual systems that may need them. As shown in Figure 4, the ANSI three schema approach, once the target stage is reached, the target environment should consist of conceptual data models which serve as the data architecture for physically shared data accessed by multiple applications, represented by external user views.

A significant number of the legacy systems were developed without conceptual or logical data models -- models now needed to support data standardization. Reverse engineering is the technique used to extract logical data models from a legacy system. As the universe of legacy systems has been narrowed by selection of migration systems, the quantity of work required to support data standardization has been drastically reduced. Figure 5 shows this concept of focusing on deriving data models from migration systems to support data standardization. As shown, the majority of the data requirements are derived, modeled, and standardized from the designated migration systems. If the systems are properly selected, the majority of the required functionality and data requirements should be derived from the migration system. The data requirements not contained in the migration systems need to be identified and extracted from the remaining systems. These requirements are indicated by scattered dots in the legacy environment.

Data assets are more than data elements (i.e., physical data structures). Data assets also include business data requirements represented, as data models linked to physical data structures. The models must represent the policies, strategies and tactics of organizational operation. Model creation steps include identification, refinement, validation, and linking of all business functions, policies, rules, and activities. Linking data items ensures model contents are supported by physical evidence.

Figure 6 illustrates the various activities identified as necessary to achieve DoD-wide enterprise information systems. For the functional areas represented on the left, a set of legacy systems currently satisfies component-specific operational requirements rather than DoD wide strategic requirements. As an interim step towards DoD enterprise systems, functional area steering committees are in the process of designating migration systems for business domains. Again, migration systems are representative of a group or class of legacy systems having the same, similar or overlapping information and/or domain functionality. Migration systems must be capable of satisfying DoD wide requirements, provide existing (i..e., as-is) functionality, and capturing data from the replaced legacy systems.

Functional area working groups are currently conducting activities to define to-be business process and logical data models with the ultimate goals of reduced cost and increased efficiency. These models become the baseline defining the operational requirements of to-be DoD enterprise information systems. Selected migration system data assets will be migrated, some enhanced, into these to-be systems.

Currently, there is no direct mapping between data elements and organizational business rules, business domain information, system functional requirements, functional dependencies, and organizational data distribution architectures. As-is data elements and their embedded business requirements are often in conflict with or are insufficient to satisfy the to-be business requirements. Reverse engineering the as-is migration systems is essential to recovering the associated business requirements at the operational, tactical, and strategic levels.

CUSTOMER	CONTRACTOR ISSUES REPRESENTATIVES	PERFORMANCE TEAM	FUNCTIONAL PROPONENT	SYSTEM IMPLEMENTERS
Office of the Director of Defense Information Deputy Director DD Representative	**DoD Contract Holder** Contract Officer (CO) Contracting Officer Rep (COR) Contracting Officer Tech Rep (COTR) **ISSAA** Govt. Legal rep **Reverse Engineered System Contractor's Office/Legal Dept.** Contractor's VP Legal Dept. **Contractor Contract Office/Legal Contracting Office** Legal Advisor	DISA/CIM/DAPMO Project Manager Lead Engineer Functional Liaison **HITC** Program Manager Project Area Leader Lead Engineer	**Program Office of the reverse engineered system** PM System POC Database POC Deputy PM **DoD Functional Area** Data Administrator Planning Division Strategic Planner	**Contractor holding Development Contract for reverse engineered system** Program Manager Product Dev Lead Engineer 5 Members of Tech Staff **FFRC** Lead Engineer Group Leader 2 Engineers

Figure 2: Representative Client Maze

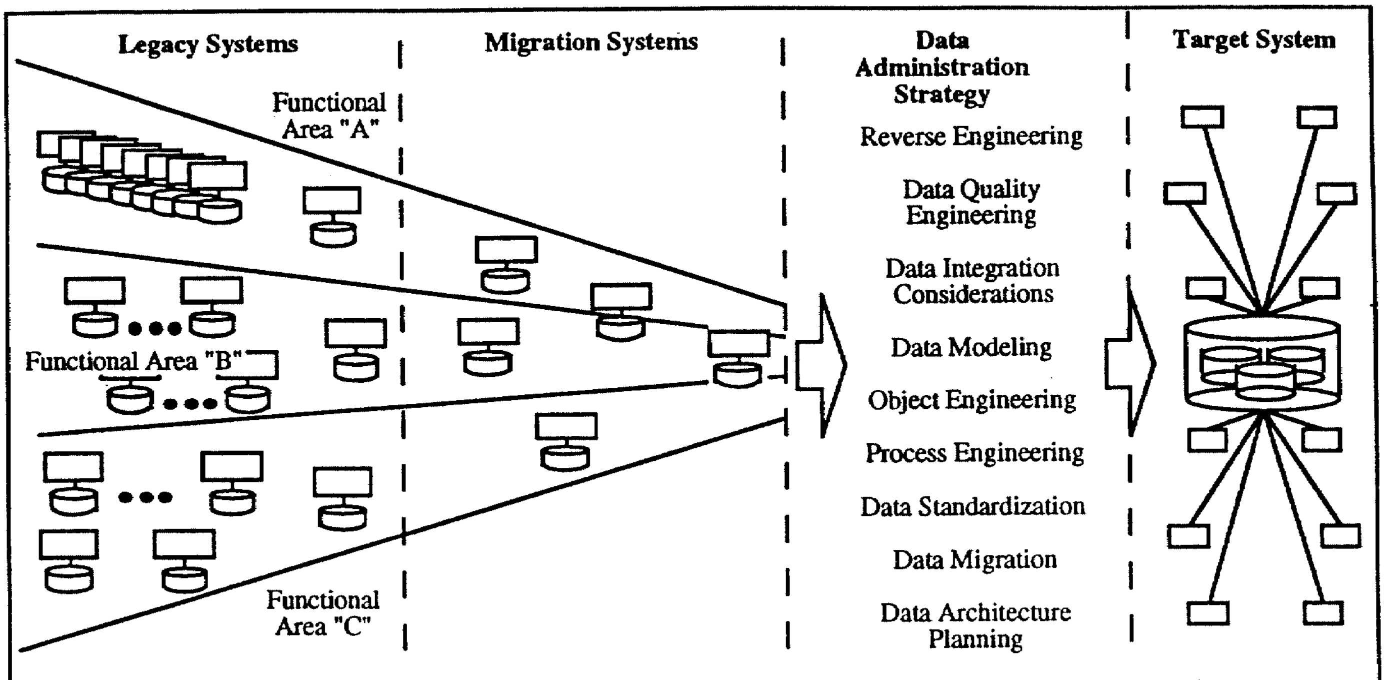


Figure 3. The Big Picture: Evolution towards a Target System Environment.

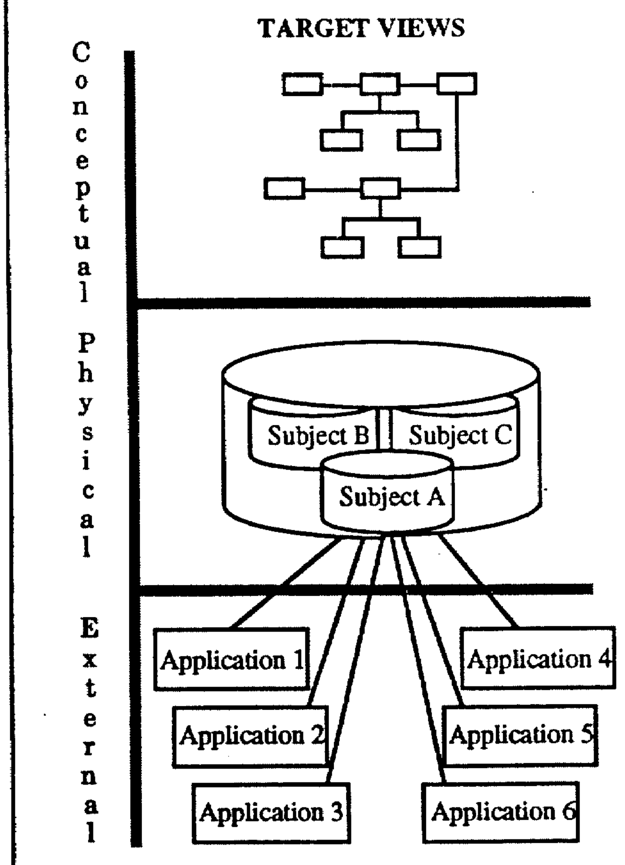


Figure 4. Three Schema View Of Target Environment.

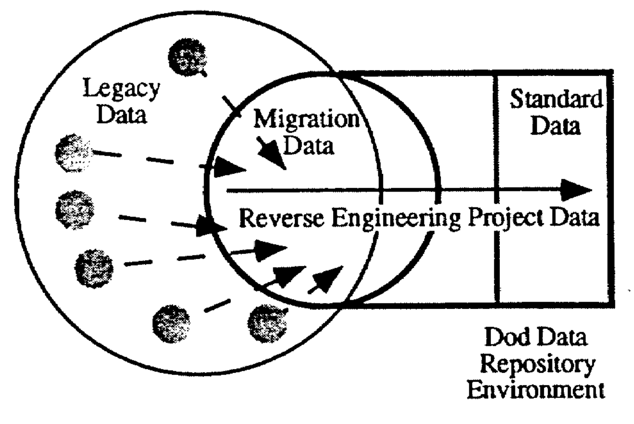


Figure 5. Systems and data migration to support data standardization

Recovered as-is requirements are then compared against the to-be business requirements to identify business requirement gaps during forward engineering. In addition to identifying business requirement gaps, technological gaps are also crucial to determining if as-is migration systems satisfy system operational performance requirements and maintenance cost constraints. The system architecture of the migration systems are evaluated against technical requirements to identify technical requirement gaps. As part of forward engineering, the business requirements gap, technical requirements gaps, and data element quality are evaluated to determine the "migratability" and "integrability" of migration systems forming the basis of economic justification to forward engineer individual systems.

The DPSM program supports the integrated analysis and re-design/development activities required to modernize the selected migration systems. Due to the massive and complex nature of the baseline migration systems in DoD, modernization must be conducted in multiple phases. The initial phase (DPSM Phase I) will validate and refine the approach that has been developed to perform the re-engineering (reverse and forward engineering) analysis of six selected DoD information systems in three functional areas: Personnel, Finance, and Health Affairs. The overall thrust of the effort is to identify cross-functional "human being" related data elements within DoD for integration purposes.

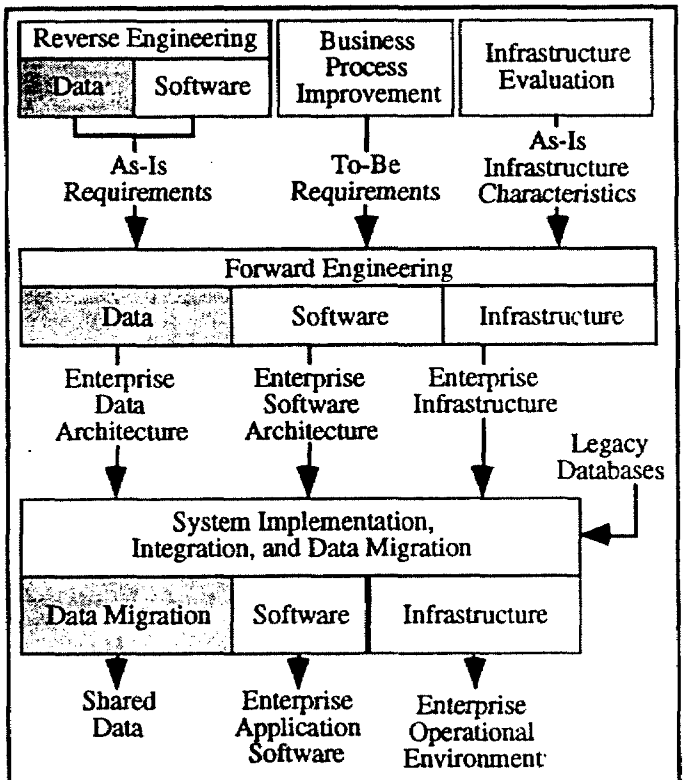


Figure 6. Scope of Reverse Engineering Efforts.

As shown in Figure 7, the Defense Civilian Personnel Data System (DCPDS) is in Personnel, the Defense Civilian Payroll System (DCPS) is in Finance, the Marine Corps Total Force System (MCTFS) is cross functional between Personneland Finance, the Composite Health Care System (CHCS), Coordinated Care Performance (CCP), and Medical Expense and Performance Reporting System (MEPRS) are all in Health Affairs. The systems currently serve major elements of the Departments of Army, Navy, Air Force, Marines, etc. Before starting this program, quantitative measures of system size and complexity were unknown. This was one of the reasons for running this initial project phase as a prototype effort before full-scale re-engineering efforts begin. The results of this phase will be used as input to economic analyses to determine selection and economic evaluation criteria of future reverse engineering candidate systems in other functional areas.

During DPSM Phase I, an important goal is validation and refinement of the reverse engineering approach. However, the primary focus is on derivation of normalized logical data models and standardization of data element names for generic elements (attributes) and prime elements (entities). These logical data models represent organizational business rules, business domain information, system functional requirements, functional dependencies, and system data architectures of the selected reverse engineered system. Re-engineering prototypes are being developed for each application. In addition, we are performing a limited cross-functional integration for a selected business domain (Calculate Pay) relevant to civilian personnel and civilian pay functional areas. In addition to the cross-system integration described above, specific objectives for DPSM Phase I are:

- requirements analyses, re-engineering analyses and design, rapid prototyping and testing of systems and subsystems, consideration of data migration and integration issues;
- an extendible inventory of legacy system data assets (data in the asset repository should be accessible via a programming interface);
- a realistic and extendible approach for reverse engineering a specified set of systems;
- identification of automated tools and requirements for tools for reverse engineering legacy systems;

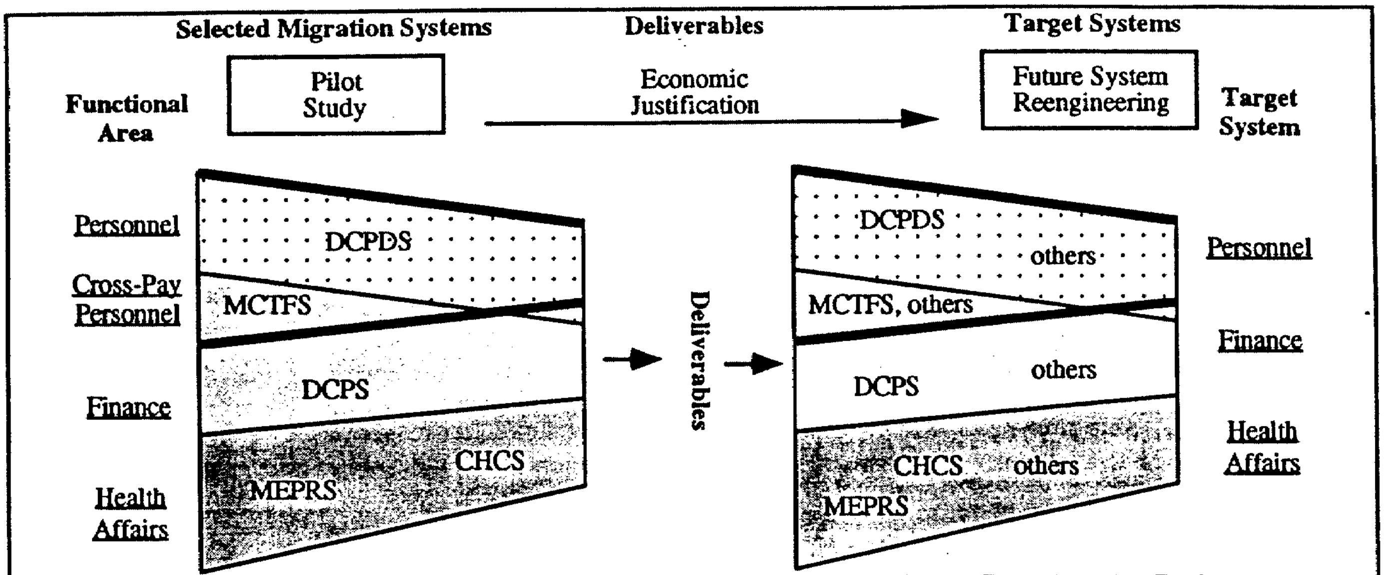
The effort also supports the extension of the DoD data model which will feed the DoD Data Repository, in a format consistent with data standardization procedures, as prescribed by DoD corporate information management strategy.

Conceptual Representation of the Re-Engineering Project Approach

Two of the systems will be used in this example. The first is the DCPDS, originally developed to provide civilian (as opposed to military) personnel support for the U.S. Air Force and which now serves more than 130 organizational customers within and outside DoD. The second is the DCPS, originally developed to pay civilians within a major organizational element of the Navy. Figure 8 presents the approach that will be used to isolate and normalize the data that is used in the "calculate pay" function.

Starting with DCPDS, the first step {1(a)} is to identify the functional domain in the system which contains the personnel data elements required to support calculate pay in DCPS. The system functions meeting domain requirements include: (a) staffing (affirmative employment), (b) job classification, and (c) employee management relations. The second step {1(b)} is to identify the functional domain of DCPS containing pay data elements supporting calculate pay within DCPS. The third step {1(c)} is to compare the results of the first step {1(a)} and second step {1(b)} to identify shared data elements used to calculate pay in DCPS forming the overlapping segment identified as {1(c)}.

If additional systems contain domains which contribute to calculate pay or perform the same function themselves they can be compared in pair-wise fashion iteratively until common data are isolated. (For an example of the related military domains see Figure 13.) In multiple systems comparisons, more extensive analysis is required because business domains, business rules, functional requirements, and functional dependencies must be cross compared at each stage to isolate not shared but "identical" data elements and structures.



Tasks
- Reverse Engineering Software
- Reengineering Data Structure
- Technology Insertion Opportunity Study
- Data Migration Approach
- System Survey
- DPSM Process Methodology

Deliverables
- High Level Process Model
- Logical Data Model
- Traceability Matrix
- Standardized Data Elements
- Technology Insertion Recommendations
- Data Migration Plan
- System Survey
- Process, Methodology, Tools & Usage Reports

Future Reengineering Projects
- Reusable Software Requirements
- Integrated DoD Enterprise Database
  - DoD Enterprise Data Model
  - DoD Standard Data Elements
- DoD Enterprise Infrastructure
- System Migration Plan
- System Complexity Matrix
- Economic Justification
- Reengineering Methodology & Toolset

Figure 7. Migration Systems Context.

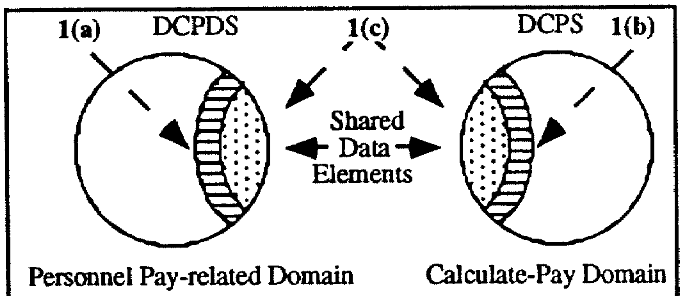


Figure 8. Domain Relationship for Cross System Integration for Two Systems.

Framework Overview, Modeling Approach, and Model Management

The framework depicted in Figures 9-10 outlines the steps for reverse engineering selected systems. The derived logical data model associates business rules, business domain information, system functional requirements, functional dependencies, and organizational data distribution architectures to data elements. One of essential outcomes of reverse engineering is a traceability matrix linking the data model components to the physical evidence supporting their existence. The traceability matrix is critical for validating the correctness of derived logical models from physical evidence. Later the logical data models can be used by CASE tools to automatically create table structures on the selected DBMS, and the traceability matrix is used to download data from the legacy system to the appropriate tables in the DoD enterprise system.

Based on the characteristics of the systems we encountered, the key aspects of the technical approaches are: (1) divide-and-conquer; (2) extraction of business rules from software and data structures; (3) model management; (4) configuration management; and (5) schema integration. Since these systems are all large scale information systems, they usually contain several million lines of code, and thousands of data structures. Although software may be divided into modules and software modules may be used to implement various software functional requirements, a business function may be implemented as part of several different software functions. As part of the physical implementation, data structures also have multiple roles. Data structures directly extracted from software may be transient data variables used to store processing results from stored data elements. Processing may be an implementation of business rules or low level mathematical algorithms (e.g., sort). Data structures can be used to hold data presented as on-line screens or reports. Data structures can also be used to access data elements stored in databases or files. Hence, data structures defined in software and data dictionaries, if they exist, may represent conceptual, external, or physical views of data elements.

The most difficult aspect of reverse engineering is to discover business rules and data entities from software and data structures. Massive quantities of data structures and associated code, force a divide-and-conquer archeological approach to discovering data elements and organizing them into categories. Our approach is top-down then bottom-up. During the top-down step, we analyze material relevant to the conceptual view (e.g., user screens, reports, policy statements). This helps to establish draft high level "as-is" business process and data model frameworks. These are used to quickly identify a set of conceptual buckets for "holding" relevant categories of data structures for a specific domain. We use the business process model to divide the business data model into views.

Each data model view corresponds to a business process. The functional dependencies between views are inherited from the relationships between processes in the high level process model. We also identify the non-transient data structures accessed (created, inserted, updated, and deleted) aspects of processes. We then derive more detailed logical data models and link these data structures to the derived data entities using the traceability matrix. All data structures and data entities are linked to their associated operational, tactical, and/or strategic requirements.

For transient data structures computed from non-transient data elements and later used for updating other non-transient data, we define structure entities to capture the business rules associating these non-transient data elements.

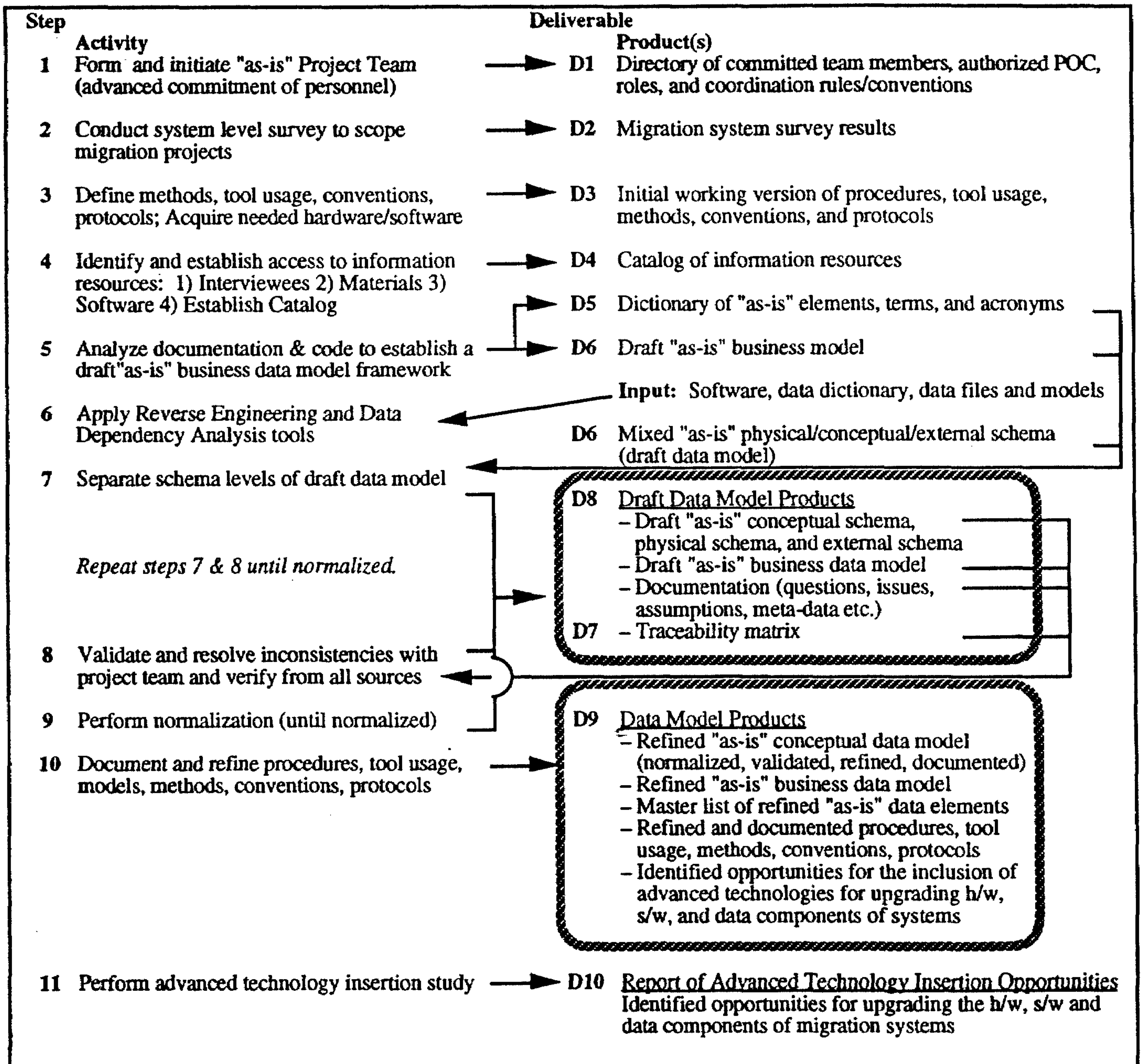


Figure 9. Reverse Engineering Framework (1 of 2).

As shown in Figure 11, each version of a reverse-engineered data model is associated with an encyclopedia. Our configuration management approach defines standardized policies and procedures making it feasible for DPSM team members to review each other's work and understand information in other project encyclopedia/directory structures. The encyclopedias store information in three separate dictionaries. The Plan Dictionary contains planning information, the Data Dictionary contains the logical data models and associated information, and the Design Dictionary contains physical structures and related information.

To enhance traceability, physical evidence obtained is also catalogued in an Information Resource Catalogue (IRC) database. The IRC contains sources of information relevant to each reverse-engineered system including system manuals, source code, directives, interview results, etc. It provides an electronic index for the information resources gathered during the reverse-engineering life cycle. The information resources are physically stored in filing cabinets. The traceability matrix is used to identify and/or trace the correlation of items contained in the various models and document the satisfaction of business requirements and rules. The data models developed using the IE: Advantage CASE tool may be imported to the IRC. The traceability matrices stored in the encyclopedia are also loaded into the IRC. The IRC also permits users to link physical evidence to the data model with an interface that users can use to query the contents of IRC, the data modeling status, and linkage between physical elements, logical entities, and business requirements and rules.

Lessons learned From Experience with the Framework

By the time this paper is presented we will have completed re-engineering the systems described previously and will be well into the next set of reverse engineering analysis projects for the Department. At the time of writing the framework is functioning as expected in that we are producing quality models in relatively short periods of time.

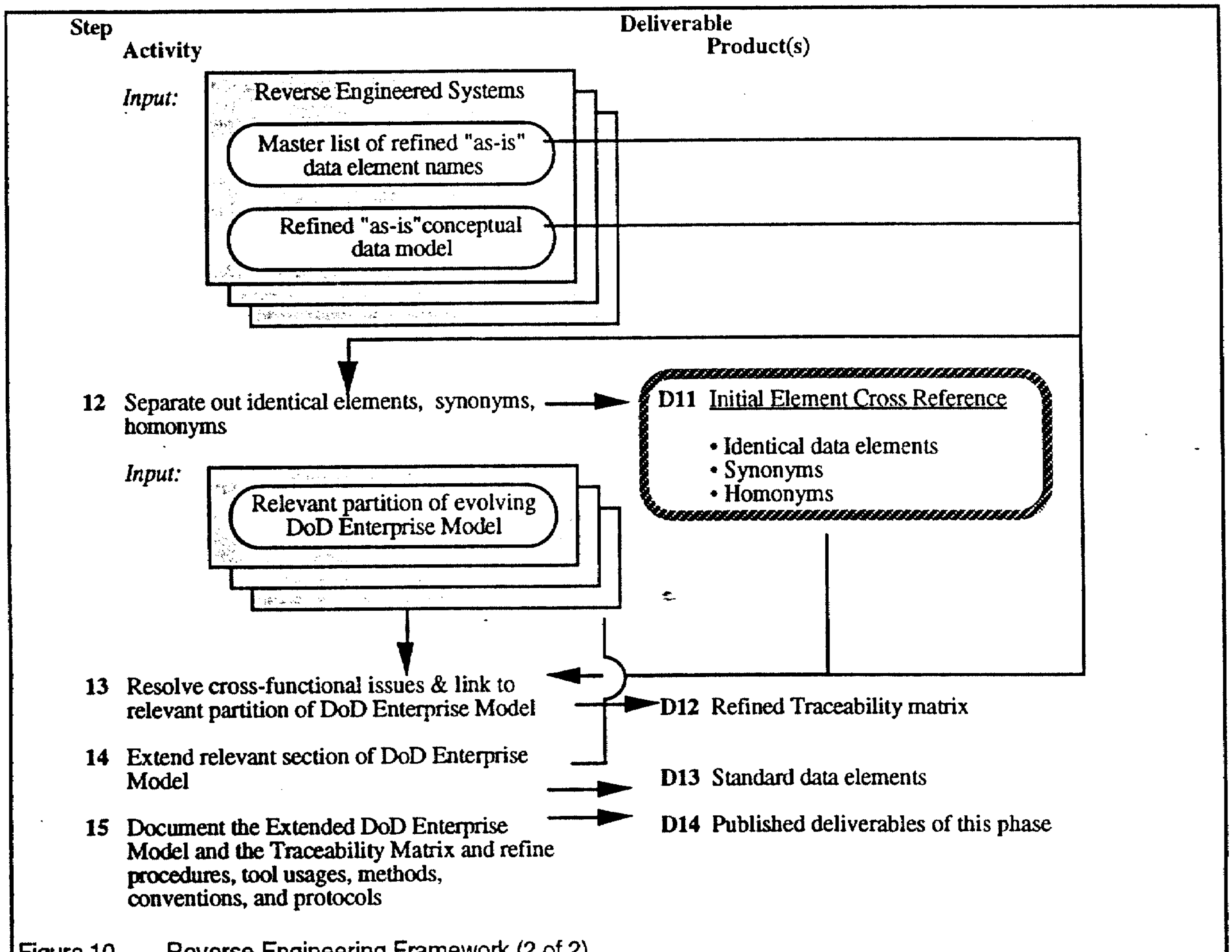


Figure 10. Reverse Engineering Framework (2 of 2)

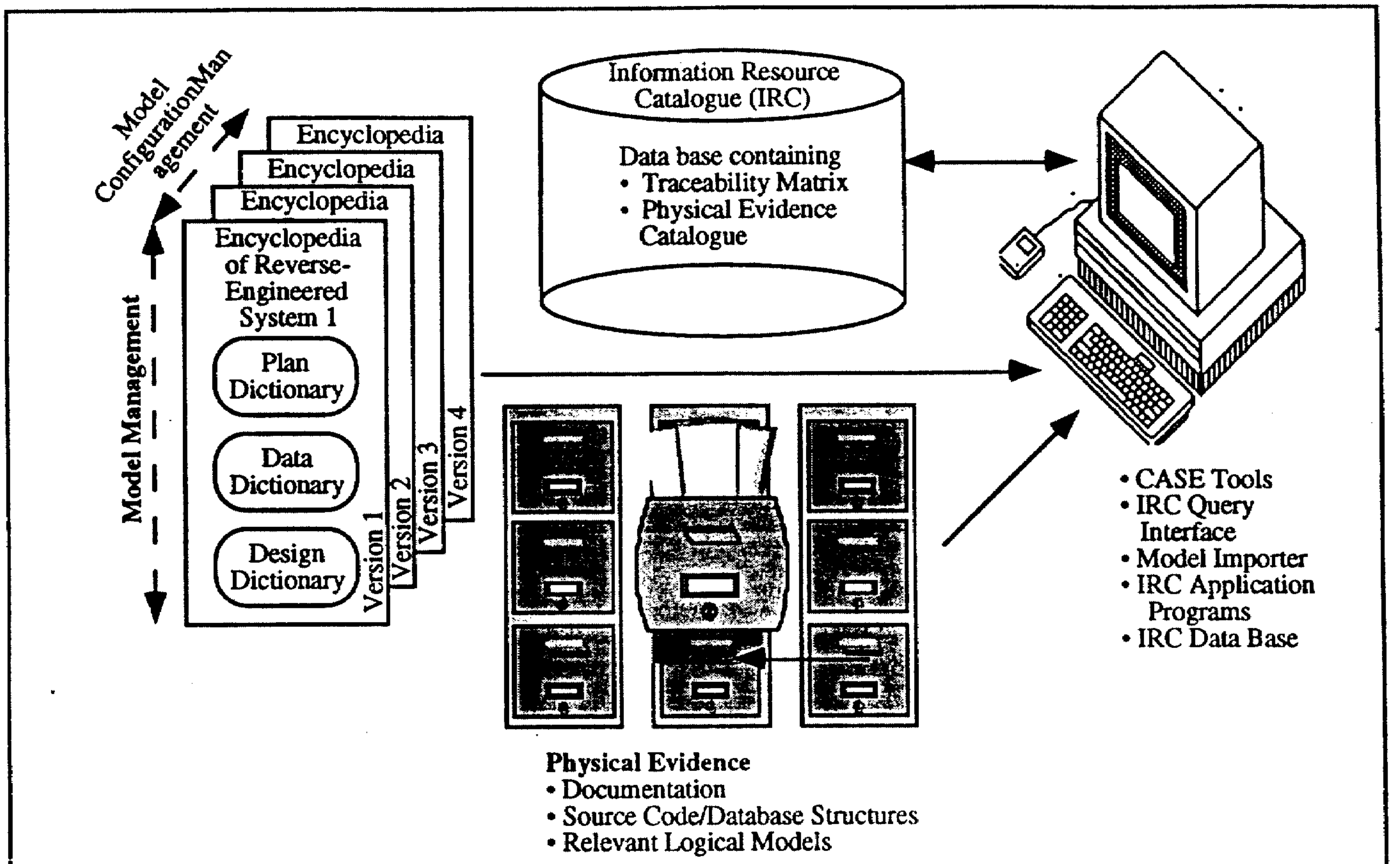


Figure 11. DPSM Modeling Approach and Model Management.

Since resources were limited, the project was focused based on the 80/20 rule -- do not try to be perfect; try to get at 80 percent of the requirements represented by 20 percent of the available resources (especially the systems and functional experts). During the course of DPSM Phase I, we have learned several important lessons. Most importantly, the value of the reverse engineering products has been consistently under estimated by management. Secondly, the costs of the reverse engineering efforts are very difficult to estimate up front. Third, and though it sounds elementary, we discovered the value of getting formal commitment and authorization from the major parties involved.

Products of Reverse Engineering

Educating management to the value and potential use of revere engineering products has been a challenge. The general perception seems to be the result of one of the primary points we hope to make with this paper -- reverse engineering is a substantially broader and more complex task than just "restructuring the code" [1, 2] -- is a tedious but necessary chore to perform before moving on to more glamorous tasks such as defining data architectures. The value of products developed during the reverse engineering analysis is consistently underestimated by management. The general perception seems to be the result of one of the primary points we hope to make with this paper -- reverse engineering is a substantially broader and more complex task than just restructuring the code. We believe this short sighted view is reflected in the current crop of CASE tools touted as reverse engineering solutions. Typical reverse engineering tools only perform limited pieces of the actual work required. Even if the functional and technical experts help the reverse engineers clean up the analysis products, the products do not include for instance the links to physical evidence provided in the traceability matrix and model management support described here as services offered by the framework. More importantly, using such tools in isolation may be as damaging as performing inadequate, inaccurate, or incomplete systems or software requirements engineering. A brief examination of the products of the reverse engineering analysis offered by the framework shows their relative importance. Several of our reverse engineering products, although not typically considered in reverse engineering contexts, provide extensive benefits to understanding and documenting the current baseline, supporting forward engineering, and standardizing and structuring data as shown in Figure 12. By briefly examining them we hope to show how our framework generates the basis for key system components used as inputs later in the development life cycle. In addition, our specific DPSM Phase I project context feeds data standardization efforts occurring at a level above the context of each specific project. These aspects are discussed below.

Outputs of Reverse Engineering	Functional Economic Analysis	Forward Engineering	Defense Data Repository System	Technical Infrastructure Evaluation	Business Process Improvement	Configuration Management	Cross functional model integration	Data Architecture	Data Migration Planning Activities	Data Standardization Activities	Data Quality Assurance Activities	Data Migration Activities	Object Engineering
High level (initial) process models		X	X		X		X	X	X	X	X	X	X
Logical data models		X	X				X	X	X	X	X	X	X
Standardized data elements			X				X	X	X	X	X	X	
Traceability matrix		X	X			X	X	X	X			X	X
Inputs to data migration plans	X	X			X			X	X				X
System survey information	X	X				X							
Process, methodology, tools	X	X	X			X	X				X		
Tool usage reports	X	X	X				X		X				
Technology insertion opportunity recommendations	X	X		X									

Figure 12. Uses of Reverse Engineering Products

Value of Reverse Engineering Products

The specific outputs from each individual reverse engineering effort encompass more than just data models. In addition to logical data models and standard data elements, the specific outputs of the reverse engineering framework include the following.

- **High Level Model View Decomposition Hierarchies** - These are used to size the system, scope the project, and define the logical data model views.
- **Traceability Matrix** - The matrix is contained in the IRC (described previously) serves as a link between the physical evidence and the logical data entities. The matrix supports existence of the data model components with statutes, regulations, policy guidance, etc.
- **Technology Insertion Recommendations** - As long as these well-qualified teams are analyzing legacy information systems, it seems entirely appropriate to take note of major areas where technology insertion recommendations would be useful components in the resulting migration plans. Areas such as advanced database and communications network technologies are typical recommendations.
- **Process, Methodology, and Tool Usage Reports** - Since none of the DPSM Phase I projects are scheduled to produce complete reverse engineering analyses of any system, all of these feed into an overall economic justification of the time and effort required to complete the reverse and forward engineering of the specific systems.
- **System & Data Migration Plans** - IE concepts are often criticized because they prescribe solutions for situations where much of the work is new development but are not useful for dealing with large installed bases of legacy systems. The systems migration plan prescribes the necessary steps to make the existing legacy system compliant with IE concepts based on guidance from existing directives [3, 4]. It is a plan for bridging the gap between the "as is" legacy information system serving a narrow purpose and the "to be" integrated Department-wide information system.

Products Contributing to DoD Data Standardization Efforts

At a somewhat higher level the combined output from the DPSM Phase I deliverables will feed a number of meta-projects also occurring within DAPMO. These include the following.

- **Re-engineering Framework and Tool Set** - As we have stated, we are continually revising the framework to reflect our experience. Both it and the tool set will be used for at least the next decade within DoD as it moves toward enterprise information systems.

- **Reusable Software Requirements** - Another by product will be reusable software requirements in the form of data models, IRC domain specific rule sets, and perhaps eventually, the software constructed to support the requirements.
- **DoD Enterprise Data Model, DoD Standard Data Elements, Integrated DoD Enterprise Database** - These are all eventual goals of the DAPMO. All outputs from the reverse engineering projects are integrated with these Department-wide efforts.

The Need for Pre-planning System Scoping Surveys

Costs of the re-engineering efforts are difficult to estimate. One vendor with whom we are familiar charges a flat $1.00/line of COBOL code in the system. For the DPSM Phase I systems we were unable to even estimate the lines of code in the medical portion because of the unstructured nature of the MUMPS programming language code. One measure of prowess among MUMPS programmers is how complex a program can be written with a single line of code. Perhaps this accounts for the various estimates in the number of lines of MUMPS code in CHCS ranging between 1.3 million to 2.5 million depending on whom you ask.

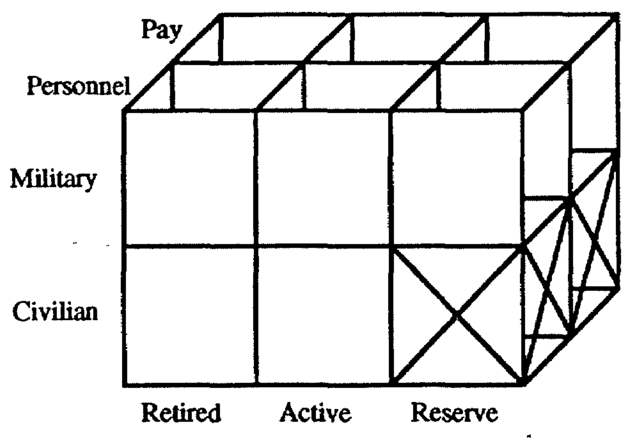


Figure 13. Dimensions of Integration of Pay and Personnel Functional Areas.

One of the major deliverables of DPSM Phase I is "an integrated civilian pay and personnel model." It turns out there are several dimensions of potential integration shown below in Figure 13. Developing an integrated model of these two functional areas has been a non-trivial task in itself without adding additional complications for military vs. civilian and active vs. retired. Convincing management of the careful sequence of steps required to develop a robust and accurate model was a major task. As a result of our experiences, we now insist on a pre-project system survey intended to size the system and scope the project prior to sizing and costing reverse engineering projects.

Implications

Development activies: Space does not permit a full discussion of all the framework implications shown in Figure 14, however, our experience has demonstrated that reverse engineering contributes to virtually all development activities. This implies that further exploration of all of these relationships should be considered as DoD modernizes this program. At the very least all future information systems development should be considered in the broader context of information engineering and data administration activities.

Technical vs. administrative issues: We mentioned previously that work progressed on DPSM Phase I as expected. Although administrative delays occurred in most cases prior to project team formation (Step 0), once projects were started, work proceeded rapidly primarily due to high, end user buy-in and participation. Our assessment is that the technological issues seem manageable - administrative issues have been the biggest obstacles to on-time delivery of reverse engineering analyses. The previously mentioned formal project authorization has been helpful in reducing the impact of these administrative delays. However, it must be emphasized that authorizations are required by all affected parties. For instance, a typical project requires functional and technical representation by the best resources an organization has. Both the functional and technical organization providing the expertise must provide authorization. And in cross-system integrated reverse engineering views, the major organizations providing experts and resources for both systems must provide authorization. Our experience is that this problem can rather quickly become a difficult coordination process not just for commitment and support but also for providing status and briefings if the team representatives are not fully empowered.

Tool Assessment Relative to I-CASE: Finally, we believe that the current focus of reverse engineering CASE tools and CASE tool development efforts are concentrating on code analysis. While it is true that most organizations will have a more homogeneous information systems base than the Department of Defense, it is also true that much of the remainder of the federal government will have configurations similar to the Department of Defense and will be unable to prescribe a single CASE tool solution for reverse engineering. A further criticism of the reverse engineering CASE tools is the focus on code analysis with little or no assistance for processes such as extracting business rules. We anticipate our work will be able to define a rudimentary set of requirements for reverse engineering CASE tools operating in a heterogeneous environment.

Keywords

Design, reverse engineering, software and system requirements and specifications, data architecture, business rules, data modeling

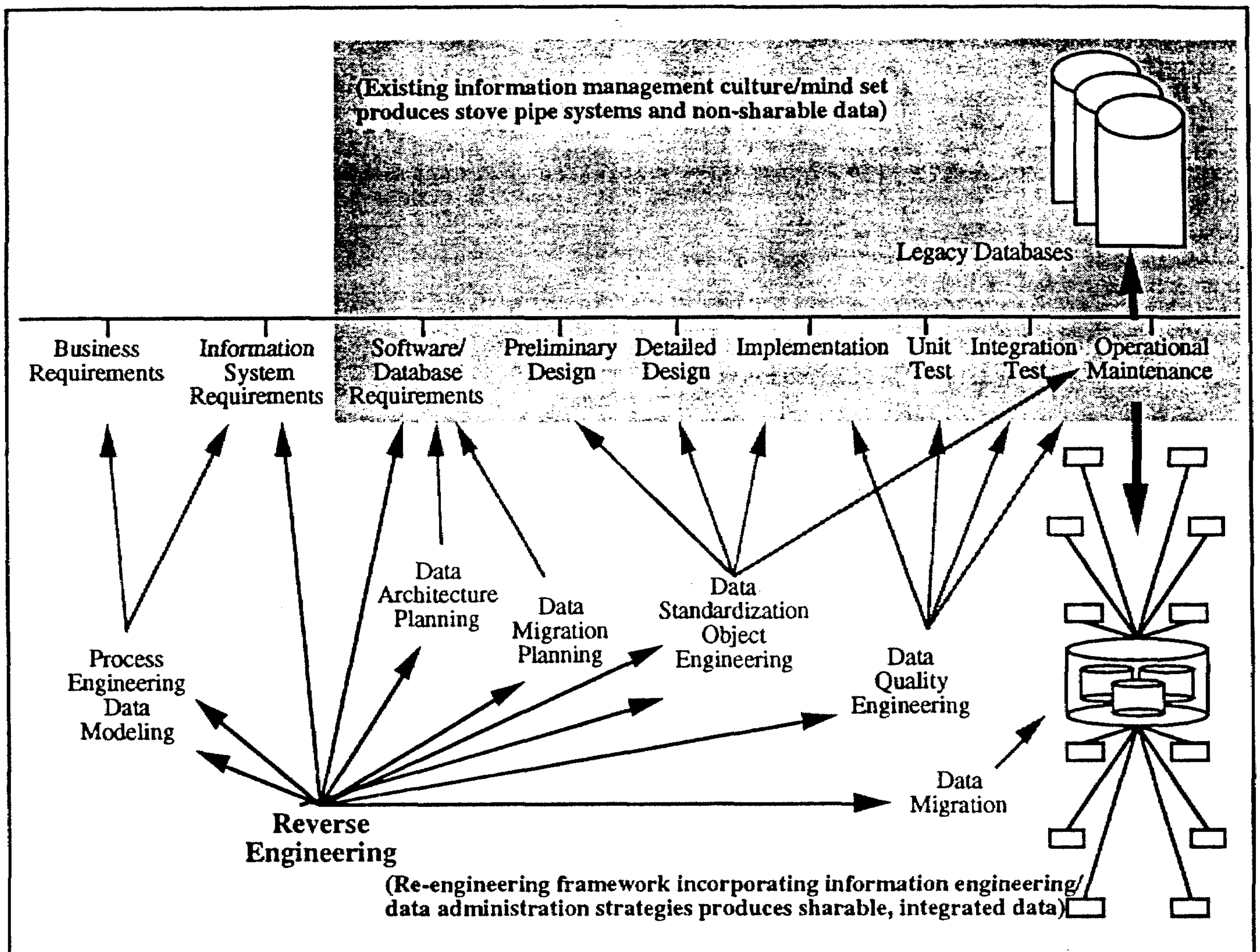


Figure 14. Reverse Engineering Contributions to Development Activities.

Acknowledgements

None of these reverse engineering projects could have achieved the real, tangible results they have without the co-operation of the respective functional and technical communities and in particular the Civilian Personnel, Civilian Pay, Health Affairs, and Procurement communities. We acknowledge their participation in making DPSM Phase I a success. This paper benefited from comments and suggestions made by the Defense Finance and Accounting Data Administration Group.

References

[1] V. R. Basili and H. D. Mills "Understanding and Documenting Program" IEEE Transactions on Software Engineering May 1982. SE-8(3):270-283

[2] Elliot Chikofski and James H. Cross II "Reverse Engineering and Design Recovery: A Taxonomy" *IEEE Software* January 1990 7(1):13-17.

[3] DoD Directive 8020.1-M *Functional Process Improvement (for Implementing the Information Management Program of the Department of Defense,* August 1992 (Draft).

[4] DoD Directive 8320.1-M *Data Standardization,* August 1992 (Draft).

[5] Richard E. Fairley *Software Engineering Concepts,* New York: McGraw-Hill Book Company, 1985.

[6] Clive Finkelstein *Information Engineering: Strategic Systems Development,* Addision Wesley Publishing Company, 1993.

[7] Donald C. Gause and Gerald M. Weinberg *Exploring Requirements: Quality Before Design,* Dorset House Publishing, New York, 1989.

[8] Richard Nolan "Managing the Crisis in Data Processing" *Harvard Business Review,* 1979.

[9] David Sharon "A Reverse Engineering and Re-Engineering Tool Classification Scheme" *Reverse Engineering Newsletter* Number 4 January 1993.

[10] Chris Staiti and Paul Pinella "DoD's Strassmann: The Politics of Downsizing" *Datamation* October 15, 1992, pp. 107-110.

[11] *Status of the Department of Defense Corporate Information Management (CIM) Initiative,* October 1992 - reprint edition.